Pre-training Models

Xu Tan
Microsoft Research Asia
xuta@microsoft.com
Outline

• Overview of pre-training
• Taxonomy of pre-training: context vs contrast
• More discussion about pre-training
• Summary
Pre-training

• What is pre-training?
  • The training in advance of standard training

• Why pre-training?
  • The standard training (data/model size) is not enough

• What to learn in pre-training?
  • Representation learning: more general, self-supervised
  • Task learning: more task specific, supervised

• When and where to apply pre-training?
  • Any tasks that data/model size are not enough
  • NLP, CV, Speech, and more

• How to learn in pre-training?
Pre-training in NLP

• Pre-training + Fine-tuning, a new paradigm of NLP
Pre-training in NLP

- **Word2Vec** (Mikolov et al., 2013)
- **CoVe** (McCann et al., 2017)
- **ULMFiT** (Howard and Ruder, 2017)
- **ELMo** (Peters et al., 2018)
- **OpenAI GPT** (Radford et al., 2018)
- **OpenAI GPT-2** (Radford et al., 2019)
- **BERT** (Devlin et al., 2018)
- **XLM** (Lample and Conneau, 2019)
- **MT-DNN** (Liu et al., 2019)
- **UNILM** (Dong et al., 2019)
- **MASS** (Song et al., 2019)
- **XLNet** (Yang et al., 2019)

...
Figure 1: Exponential growth of number of parameters in DL models
Pre-training in NLP

Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq

ULMFiT
Multi-lingual

ELMo
Transformer

MultiFiT
Cross-lingual

Bidirectional LM

GPT
Larger model
More data

GPT-2
Defense

Grover

ERNE (Baidu)

BERI-wm

VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unico-der-VL
LXMERT
VL-BERT
UNITER

ERNE (Tsinghua)

KnowBert

SpanBERT
RoBERTa

XLM
UDify

MT-DNN
Knowledge distillation

MASS
UniLM

Span prediction
Remove NSP
Longer time
Remove NSP
More data

MT-DNN
KD

XLNet
Pre-training in CV——Self-supervised

Top 1 accuracy on ImageNet with self-supervised pre-training
### Pre-training in CV — Supervised/Semi-supervised

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>Top 1 Accuracy</th>
<th>Top 5 Accuracy</th>
<th>Number of params</th>
<th>Extra Training Data</th>
<th>Code</th>
<th>Result</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Meta Pseudo Labels (EfficientNet-L2)</td>
<td>90.2%</td>
<td>98.8%</td>
<td>480M</td>
<td>✓</td>
<td></td>
<td></td>
<td>2021</td>
</tr>
<tr>
<td>2</td>
<td>Meta Pseudo Labels (EfficientNet-B6-Wide)</td>
<td>90%</td>
<td>98.7%</td>
<td>390M</td>
<td>✓</td>
<td></td>
<td></td>
<td>2021</td>
</tr>
<tr>
<td>3</td>
<td>NFNet-F4+</td>
<td>89.2%</td>
<td>527M</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>2021</td>
</tr>
<tr>
<td>4</td>
<td>ALIGN (EfficientNet-L2)</td>
<td>88.64%</td>
<td>98.67%</td>
<td>480M</td>
<td>✓</td>
<td></td>
<td></td>
<td>2021</td>
</tr>
<tr>
<td>5</td>
<td>EfficientNet-L2-475 (SAM)</td>
<td>88.61%</td>
<td>480M</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>2020</td>
</tr>
<tr>
<td>6</td>
<td>ViT-H/14</td>
<td>88.55%</td>
<td>632M</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>2020</td>
</tr>
<tr>
<td>7</td>
<td>FixEfficientNet-L2</td>
<td>88.5%</td>
<td>98.7%</td>
<td>480M</td>
<td>✓</td>
<td></td>
<td></td>
<td>2020</td>
</tr>
<tr>
<td>8</td>
<td>NoisyStudent (EfficientNet-L2)</td>
<td>88.4%</td>
<td>98.7%</td>
<td>480M</td>
<td>✓</td>
<td></td>
<td></td>
<td>2020</td>
</tr>
<tr>
<td>9</td>
<td>Mixer-H/14 (JFT 300M pre-train)</td>
<td>87.94%</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2021</td>
</tr>
<tr>
<td>10</td>
<td>ViT-L/16</td>
<td>87.76%</td>
<td>307M</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>2020</td>
</tr>
</tbody>
</table>

Top 1 accuracy on ImageNet with supervised/semi-supervised pre-training
Pre-training in Speech

SOTA WER on LibriSpeech with self-supervised and semi-supervised training
Pre-training in Speech

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>Word Error Rate (WER)</th>
<th>Extra Training Data</th>
<th>Code</th>
<th>Result</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conformer + Wav2vec 2.0 + SpecAugment-based Noisy Student Training with Libri-Light</td>
<td>1.4</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>2020</td>
</tr>
<tr>
<td>2</td>
<td>Conv + Transformer + wav2vec2.0 + pseudo labeling</td>
<td>1.5</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>2020</td>
</tr>
<tr>
<td>3</td>
<td>ContextNet + SpecAugment-based Noisy Student Training with Libri-Light</td>
<td>1.7</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>2020</td>
</tr>
<tr>
<td>4</td>
<td>SpeechStew (18)</td>
<td>1.7</td>
<td>×</td>
<td>-</td>
<td>-</td>
<td>2021</td>
</tr>
<tr>
<td>5</td>
<td>Multistream CNN with Self-Attentive SRU</td>
<td>1.75</td>
<td>×</td>
<td>-</td>
<td>-</td>
<td>2020</td>
</tr>
<tr>
<td>6</td>
<td>wav2vec 2.0 with Libri-Light</td>
<td>1.8</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>2020</td>
</tr>
<tr>
<td>7</td>
<td>ContextNet (L)</td>
<td>1.9</td>
<td>×</td>
<td>-</td>
<td>-</td>
<td>2020</td>
</tr>
<tr>
<td>8</td>
<td>Conformer (L)</td>
<td>1.9</td>
<td>×</td>
<td>-</td>
<td>-</td>
<td>2020</td>
</tr>
</tbody>
</table>

SOTA WER on LibriSpeech with self-supervised and semi-supervised training
How to learn in pre-training

- Learning paradigm
  - Supervised learning
  - Unsupervised learning
  - Semi-supervised learning
  - Reinforcement learning
  - Transfer learning
  - Self-supervised learning

- Pre-training
  - In this talk, we focus more on self-supervised learning
  - Context based and contrast based
Context based vs Contrast based

• Context based
  • Autoregressive Language Model (LM): ELMo [3], GPT-1/2/3 [4,5,6]
  • Denoising Auto-Encoder (DAE): MLM (BERT[7], RoBERTa[9], ERNIE[21,23], UniLM[14], XLM [15]), Seq2SeqMLM (MASS [11], T5 [17], ProphetNet [43], BART[12])
  • Permutated Language Model (PLM): XLNet [10], MPNet [27]

• Contrast based
  • Context-Instance Contrast
    • Predict Relative Position (PRP): Jigsaw, Rotation Angle [45], Sentence Order Prediction (ALBERT [19], StructBERT [20])
    • Maximize Mutual Information (MI): Deep InfoMax/InforWord [28], AMDIM [29], Contrastive Predictive Coding [30] (wav2vec [41,42]), Replaced Token Detection (word2vec [1], ELECTRA[18])
  • Context-Context Contrast
    • DeepCluster [32], CMC [31], MoCo [34,37], SimCLR [35,38], BYOL [36], Next Sentence Prediction (BERT [7])
Context based: LM

• **Language model** \( \mathcal{L}_{LM} = - \sum_{t=1}^{T} \log p(x_t|x_{<t}) \)
  
  • Natural, Joint probability estimation
  • Left to right, no bidirectional context information

• **ELMo** [3], **GPT** [4], **GPT-2** [5], **GPT-3** [6]
  
  • ELMo: NAACL 2018 best paper, GPT-3: NeurIPS 2020 best paper
  • GPT-3 has 175 billion parameters, the largest model before (1.7 Trillion, Switch Transformer [46])
Context based: DAE

• Denoising Auto-Encoder
  • DAE: Noisy Input, reconstruct whole clean input
    \[ L_{DAE} = - \sum_{t=1}^{T} \log p(x_t | \hat{x}, x_{<t}) \]
  • MLM: Noisy Input (with mask tokens), reconstruct mask tokens
    \[ L_{MLM} = - \sum_{\hat{x} \in m(x)} \log p(\hat{x} | x_{\setminus m(x)}) \]
  • Seq2SeqMLM: Noisy Input (with mask tokens), reconstruct mask tokens, with encoder-decoder framework
    \[ L_{S2SMLM} = - \sum_{t=1}^{j} \log p(x_t | \hat{x}_{\setminus x_{i:t-1}}) \]

• BERT [7], MASS [11], RoBERTa [9], XLM [15], ERNIE [21,23], UniLM [14], ProphetNet [43], T5 [17], BART [12]
Context based: MLM—―BERT

• BERT [7]: Bidirectional transformer, vs GPT [4,5,6], ELMo [3]

Context based: MLM———BERT

• BERT [7]: Bidirectional transformer

Context based: Seq2SeqMLM—MASS

• MASS: MAsked Sequence to Sequence pre-training [11]
  • MASS is carefully designed to jointly pre-train the encoder and decoder

• Mask k consecutive tokens (a sentence segment)
  • Force the decoder to attend on the source representations, i.e., encoder-decoder attention.
  • Force the encoder to extract meaningful information from the sentence.
  • Develop the decoder with the ability of language modeling.

Context based: PLM

• Permutated Language Model

\[ L_{PLM} = - \sum_{t=1}^{T} \log p(z_t|z_{<t}) \]

• A generalized autoregressive language model, random permute the sentence order, better use language model for pre-training
• Combine the advantages of LM and MLM
  • LM: only left context, MLM: bidirectional context
  • LM: conditional dependent, MLM: conditional independent

• XLNet [10], MPNet [27]

Context based: PLM — XLNet

• Key designs in XLNet
  • Autoregressive model, use permuted language model (PLM) to introduce bidirectional context
  • Two-stream self-attention to decide the position of next predicted token
  • Use Transformer-XL to incorporate long context

\[
\log P(x; \theta) = \mathbb{E}_{z \in \mathcal{Z}_n} \sum_{t=c+1}^{n} \log P(x_{z_t} | x_{z_{<t}}; \theta)
\]
Context based: PLM——XLNet

• Two-stream self-attention
  • Content stream: build content hidden, same as GPT/BERT in Transformer
  • Query stream: token prediction, use position as input to decide which token to predict

\[
\begin{align*}
 g_{z_t}^{(m)} & \leftarrow \text{Attention}(Q = g_{z_t}^{(m-1)}, KV = h_{z_{<t}}^{(m-1)}; \theta), \quad \text{(query stream: use } z_t \text{ but cannot see } x_{z_t}) \\
h_{z_t}^{(m)} & \leftarrow \text{Attention}(Q = h_{z_t}^{(m-1)}, KV = h_{z_{\leq t}}^{(m-1)}; \theta), \quad \text{(content stream: use both } z_t \text{ and } x_{z_t}).
\end{align*}
\]
Context based: PLM—–XLNet

• Transformer-XL
  • Recurrence mechanism: cache and reuse the representation of previous segment

\[
h_{z_t}^{(m)} \leftarrow \text{Attention}(Q = h_{z_t}^{(m-1)}, KV = [\tilde{h}^{(m-1)}, h_{z_{\leq t}}^{(m-1)}]; \theta)
\]

• Relative position embedding
  • Do not care the absolute position, but only relative position

\[
A_{i,j}^{\text{abs}} = \underbrace{E_{x_i}^TW_q^TW_kE_{x_j}}_{(a)} + \underbrace{E_{x_i}^TW_q^TW_kU_j}_{(b)} + \underbrace{U_i^TW_q^TW_kE_{x_j}}_{(c)} + \underbrace{U_i^TW_q^TW_kU_j}_{(d)}
\]

\[
A_{i,j}^{\text{rel}} = \underbrace{E_{x_i}^TW_q^TW_kE_{x_j}}_{(a)} + \underbrace{E_{x_i}^TW_q^TW_kR_{i-j}}_{(b)} + \underbrace{u^TW_kE_{x_j}}_{(c)} + \underbrace{v^TW_kR_{i-j}}_{(d)}
\]
Context based: PLM —— XLNet

• The advantage of XLNet (the task is sentence classification)
  • Bidirectional context, vs GPT

<table>
<thead>
<tr>
<th>Objective</th>
<th>Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM (GPT)</td>
<td>$\log P(is \mid \text{the task})$</td>
</tr>
<tr>
<td>PLM (XLNet)</td>
<td>$\log P(is \mid \text{the task}) + \log P(is \mid \text{sentence classification})$</td>
</tr>
</tbody>
</table>

• Dependency between predicted tokens, vs BERT

<table>
<thead>
<tr>
<th>Objective</th>
<th>Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLM (BERT)</td>
<td>$\log P(\text{sentence} \mid \text{the task is}) + \log P(\text{classification} \mid \text{the task is})$</td>
</tr>
<tr>
<td>PLM (XLNet)</td>
<td>$\log P(\text{sentence} \mid \text{the task is}) + \log P(\text{classification} \mid \text{the task is sentence})$</td>
</tr>
</tbody>
</table>
Context based: MLM+PLM——MPNet

• The pros and cons of BERT and XLNet
  • “the task is sentence classification”, predict token “sentence” and “classification”

<table>
<thead>
<tr>
<th>Objective</th>
<th>Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLM (BERT)</td>
<td>$\log P(\text{sentence}</td>
</tr>
<tr>
<td>PLM (XLNet)</td>
<td>$\log P(\text{sentence}</td>
</tr>
</tbody>
</table>

• Two aspects
  • Output dependency: dependency among the masked/predicted tokens
  • Input consistency: position information between pre-training and fine-tuning

<table>
<thead>
<tr>
<th></th>
<th>Output Dependency</th>
<th>Input Consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLM (BERT)</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>PLM (XLNet)</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

MPNet: inherit their advantages but avoid their limitations
Context based: MLM+PLM——MPNet

• **Autoregressive prediction** (avoid the limitation in BERT)
  • Each predicted token condition on previous predicted tokens to ensure **output dependency**

• **Position compensation** (avoid the limitation in XLNet)
  • Each predicted token can see full position information to ensure **input consistency**
Context based: MLM+PLM——MPNet

• The advantages of MPNet

<table>
<thead>
<tr>
<th>Objective</th>
<th>Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLM (BERT)</td>
<td>( \log P(\text{sentence} \mid \text{the task is [M] [M]}) + \log P(\text{classification} \mid \text{the task is [M] [M]}) )</td>
</tr>
<tr>
<td>PLM (XLNet)</td>
<td>( \log P(\text{sentence} \mid \text{the task is}) + \log P(\text{classification} \mid \text{the task is sentence}) )</td>
</tr>
<tr>
<td>MPNet</td>
<td>( \log P(\text{sentence} \mid \text{the task is [M] [M]}) + \log P(\text{classification} \mid \text{the task is sentence [M]}) )</td>
</tr>
</tbody>
</table>

• Position compensation, input consistency, vs. PLM (XLNet)
  • MPNet knows 2 tokens to predict, instead of 3 tokens like “sentence pair classification”

• Autoregressive prediction, output dependency, vs. MLM (BERT)
  • MPNet can better predict “classification” given previous token “sentence”, instead of predicting “answering” as if to predict “question answering”
Context based: MLM+PLM—-MPNet

• The advantages of MPNet

<table>
<thead>
<tr>
<th>Objective</th>
<th>Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLM (BERT)</td>
<td>( \log P(\text{sentence}</td>
</tr>
<tr>
<td>PLM (XLNet)</td>
<td>( \log P(\text{sentence}</td>
</tr>
<tr>
<td>MPNet</td>
<td>( \log P(\text{sentence}</td>
</tr>
</tbody>
</table>

• How much conditional information is used on average to predict a masked token? (assume all objectives mask and predict 15% tokens)

<table>
<thead>
<tr>
<th>Objective</th>
<th>Formulation</th>
<th>#Tokens</th>
<th>#Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLM (BERT)</td>
<td>( \sum_{t=c+1}^{n} \log P(x_{z_t}</td>
<td>x_{z_{&lt;c}}, M_{z_{&gt;c}}; \theta) )</td>
<td>85%</td>
</tr>
<tr>
<td>PLM (XLNet)</td>
<td>( \sum_{t=c+1}^{n} \log P(x_{z_t}</td>
<td>x_{z_{&lt;t}}; \theta) )</td>
<td>92.5%</td>
</tr>
<tr>
<td>MPNet</td>
<td>( \sum_{t=c+1}^{n} \log P(x_{z_t}</td>
<td>x_{z_{&lt;t}}, M_{z_{&gt;c}}; \theta) )</td>
<td>92.5%</td>
</tr>
</tbody>
</table>

MPNet uses the most information to predict tokens

Inherit their advantages
Avoid their limitations
Context based vs Contrast based

• Context based
  • Autoregressive Language Model (LM): ELMo [3], GPT-1/2/3 [4,5,6]
  • Denoising Auto-Encoder (DAE): MLM (BERT[7], RoBERTa[9], ERNIE[21,23], UniLM[14], XLM [15]), Seq2SeqMLM (MASS [11], T5 [17], ProphetNet [43], BART[12])
  • Permutated Language Model (PLM): XLNet [10], MPNet [27]

• Contrast based
  • Context-Instance Contrast
    • Predict Relative Position (PRP): Jigsaw, Rotation Angle [45], Sentence Order Prediction (ALBERT [19], StructBERT [20])
    • Maximize Mutual Information (MI): Deep InfoMax/InforWord [28], AMDIM [29], Contrastive Predictive Coding [30] (wav2vec [41,42]), Replaced Token Detection (word2vec [1], ELECTRA[18])
  • Context-Context Contrast
    • DeepCluster [32], CMC [31], MoCo [34,37], SimCLR [35,38], BYOL [36], Next Sentence Prediction (BERT [7])
Contrast based

• Basic idea: learn from contrast
  • Tell what is, and tell what is not

\[
\mathcal{L}_N = -\mathbb{E}_{x,y^+,y^-} \left[ \log \frac{\exp(s(x, y^+))}{\exp(s(x, y^+)) + \sum_{j=1}^{N-1} \exp(s(x, y_j^-))} \right]
\]

• Different contrast granularities
  • Context-Instance Contrast
  • Context-Context Contrast
Contrast based: Context-Instance Contrast

• Context-Instance Contrast: Global-local contrast: the local feature of a sample and its global context representation
  • Patches to their image, sentences to their paragraph, words to their sentence, and nodes to their neighborhoods.

  \[
  \mathcal{L}_N = -E_{x,y^+,y^-} \left[ \log \frac{\exp(s(x, y^+))}{\exp(s(x, y^+)) + \sum_{j=1}^{N-1} \exp(s(x, y_j^-))} \right]
  \]

• Predict Relative Position (PRP): Jigsaw, Rotation Angle [45], Sentence Order Prediction (ALBERT [19], StructBERT [20])

• Maximize Mutual Information (MI): Deep InfoMax/InforWord [28], AMDIM [29], Contrastive Predictive Coding [30] (wav2vec [41,42]), Replaced Token Detection (word2vec [1], ELECTRA [18])
Contrast based: Context-Instance Contrast

• Predict Relative Position (PRP)
  • Jigsaw, rotation angle, relative position [45]
Contrast based: Context-Instance Contrast

- Predict Relative Position (PRP)
  - Next Sentence Prediction (BERT [7])
  - Sentence Order Prediction (ALBERT [19], StructBERT [20])

(a) Word Structural Objective

(b) Sentence Structural Objective
Contrast based: Context-Instance Contrast

- Maximize Mutual Information (MI)
  - Deep InfoMax/InfoWord [28], AMDIM [29]
Contrast based: Context-Instance Contrast

- Maximize Mutual Information (MI)
  - Contrastive Predictive Coding [30]
Contrast based: Context-Instance Contrast

- Maximize Mutual Information (MI)
  - Wav2vec /Wav2vec 2.0 [41,42]
  
  \[ \mathcal{L}_m = - \log \frac{\exp(\text{sim}(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \sim Q_t} \exp(\text{sim}(c_t, \tilde{q})/\kappa)} \]

Diagram:
- Context representations
- Quantized representations
- Latent speech representations
- Raw waveform
- Transformer
- Masked
- CNN

2021/01/25
Xu Tan @ Microsoft Research Asia
Contrast based: Context-Instance Contrast

- Maximize Mutual Information (MI)
  - Replaced Token Detection (word2vec [1], ELECTRA [18])

\[
\mathcal{L}_{\text{Disc}}(x, \theta_D) = \mathbb{E} \left( \sum_{t=1}^{n} -\mathbb{1}(x_t^{\text{corrupt}} = x_t) \log D(x^{\text{corrupt}}, t) - \mathbb{1}(x_t^{\text{corrupt}} \neq x_t) \log (1 - D(x^{\text{corrupt}}, t)) \right)
\]

\[
\mathcal{L}_{\text{RTD}} = - \sum_{t=1}^{T} \log p(y_t|x_t^{\hat{x}})
\]
Contrast based: Context-Context Contrast

- Context-Context Contrast: the relationships between the global representations of different samples
  - Cluster-based Discrimination: DeepCluster [32]
  - Instance Discrimination: CMC [31], MoCo [34,37], SimCLR [35,38], BYOL [36]
Contrast based: Context-Context Contrast

- Cluster-based Discrimination: DeepCluster [32], Local Aggregation [33]
Contrast based: Context-Context Contrast

- Instance Discrimination: Contrastive Multiview Coding (CMC) [31]
Contrast based: Context-Context Contrast

• Instance Discrimination: Momentum Contrast for Unsupervised Visual Representation Learning (MoCo) [34,37]
Contrast based: Context-Context Contrast

- Instance Discrimination: A Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [35,38]
Contrast based: Context-Context Contrast

- Instance Discrimination: Bootstrap Your Own Latent A New Approach to Self-Supervised Learning (BYOL) [36]
Context based vs Contrast based

• Context based
  • Autoregressive Language Model (LM): ELMo [3], GPT-1/2/3 [4,5,6]
  • Denoising Auto-Encoder (DAE): MLM (BERT[7], RoBERTa[9], ERNIE[21,23], UniLM[14], XLM [15]), Seq2SeqMLM (MASS [11], T5 [17], ProphetNet [43], BART[12])
  • Permutated Language Model (PLM): XLNet [10], MPNet [27]

• Contrast based
  • Context-Instance Contrast
    • Predict Relative Position (PRP): Jigsaw, Rotation Angle [45], Sentence Order Prediction (ALBERT [19], StructBERT [20])
    • Maximize Mutual Information (MI): Deep InfoMax/InforWord [28], AMDIM [29], Contrastive Predictive Coding [30] (wav2vec [41,42]), Replaced Token Detection (word2vec [1], ELECTRA[18])
  • Context-Context Contrast
    • DeepCluster [32], CMC [31], MoCo [34,37], SimCLR [35,38], BYOL [36], Next Sentence Prediction (BERT [7])
How to use pre-training for downstream tasks?

• Choose pre-training task, model structure, data in pre-training
• In fine-tuning
  • Feature incorporation or fine-tunning
  • What to fine-tune? Embedding, partial layers, whole model
  • Different fine-tuning stages, layer-wise fine-tuning
  • Extra fine-tuning adaptors
• Reduce the gap between pre-training and fine-tuning
  • Different pre-training tasks for different downstream tasks
  • Make the data and model consistency with downstream tasks
  • Joint pre-training and fine-tuning
How to use pre-training for downstream tasks?

• Different pre-training tasks for different downstream tasks
  • CoLA: The Corpus of Linguistic Acceptability, prefer ELECTRA
  • RTE, MNLI, QNLI: prefer sentence pair pre-training, such as SOP
  • NER: prefer non-degeneration output hidden, BERT instead of ELECTRA
  • SQuAD: prefer span based prediction, span mask
  • NMT, text summarization: prefer seq2seqMLM or conditional sequence generation
How to use pre-training for downstream tasks?

- Compress the pre-trained model for practical deployment
  - Pruning: Compressing BERT [47], LayerDrop [48]
  - Quantization: Q-BERT [49], Q8BERT [50]
  - Parameter sharing: ALBERT[19]
  - Knowledge distillation: DistilBERT [51], TinyBERT [52], LightPAFF [53], BERT-PKD [54], MobileBERT [55], MiniLM [56], DynaBERT [57]
  - Neural architecture search: AdaBERT [58], NAS-BERT [59]
Comparison between pre-trained models for NLP, CV and Speech

• Model size and data size
  • Image: SimCLRv2/800M/300M images, DALL-E/12B/250M image-text pairs,
  • Speech: (Conformer + Wav2vec 2.0)/1B/60K hours speech data
  • NLP: GPT-3/175B/400B tokens → Switch Transformers/1.6 Trillion/180B tokens

• Context-based or Contrast-based?
  • Image, speech, more contrast based
  • NLP, more context-based

• Perception vs Cognition
  • Image, speech is more like perception
  • NLP is more like cognition
Summary of this course

- Overview of pre-training in NLP, CV and Speech
- Taxonomy of self-supervised based pre-training
  - Context based
  - Contrast based
- More discussion about pre-training
  - How to use for down-streaming tasks
  - Comparison between NLP, CV and Speech
Thank You!

Xu Tan
Senior Researcher @ Microsoft Research Asia
xuta@microsoft.com

https://www.microsoft.com/en-us/research/people/xuta/
Reference


Reference


Reference

[21] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: enhanced language representation with informative entities. In ACL, 2019.


Reference


Reference


Reference


